INESC, Porto at TRECVID 2007: Automatic and Interactive Video Search

Catalin Calistru cmc@inescporto.pt
Cristina Ribeiro mcr@fe.up.pt
Gabriel David gtd@fe.up.pt
Irene Rodrigues ipr@di.uevora.pt
Gustavo Laboreiro gustavo.laboreiro@gmail.com

October 22, 2007

Abstract

The INESC Porto group has participated in the search task (automatic and interactive). Our approach combines high-level features (the 39 concepts of the LSCOM-Lite set) with low-level features. We use a large set of low-level features with the intention of analysing as many facets as possible of each shot. The aggregation of large feature sets can be time consuming as it needs to be done at query time. We have developed the BitMatrix indexing method to speed up the search process. For each shot, binary signatures in the form of bit sequences are obtained in an off-line process. At query time, the query bit signature is compared to each of the shots signatures. The automatic run performs above the median, in spite of not using any classifier or any other knowledge sources except the translation of the query into LSCOM-Lite concepts.

1 Introduction

The INESC Porto group has participated in the search task (automatic and interactive). We combine high-level features (the 39 concepts of the LSCOM-Lite set) with low-level features. Generally the aggregation of large feature sets at query time is time consuming. In order to speed up the search process we handled both automatic and interactive search tasks from a database indexing perspective. The strategy is to analyse as many facets as possible for each shot and then combine them. The low-level features that we extract include som: several descriptors for color, several descriptors for texture, features derived from the analysis of image segments such as keypoints, and audio descriptors. The high-level features are the ones obtained from the common annotation process. In the following sections we describe the features, the indexing structure, the topic analysis process, the search strategy and finally some preliminary results and conclusions.

2 Features

INESC Porto has not proposed new feature extraction approaches. In the sequel we enumerate the features that were used and the methods to obtain them.

2.1 High-level Features

The high-level features come from the automatic annotation of the shots with the 39 concepts from the LSCOM-Lite set, which are the target of the feature extraction task. There are several groups that participated in the feature extraction task and each one produced up to 6 versions of annotation (runs), totalling 163 runs. Each run produces a set of at most 2000 relevant shots per concept bundled in one list. There are no degrees of relevance, only binary judgements: the concept is present or not. In order to evaluate the runs and choose a reference annotation we have aggregated all the annotations. For each shot and for each concept we have analysed the runs and determined whether the concept was found in more than 30% of the annotations. In that case we associated the concept to the shot.

2.2 Low-level Features

2.2.1 Color

The color feature is represented by several descriptors. We have used: Color Layout, Color Structure, Scalable Color and ColorMoments. ColorLayout, ColorStructure and ScalableColor are extracted with the MPEG-7 XM software, while the ColorMoments feature was provided by the City University of Hong Kong.

2.2.2 Texture

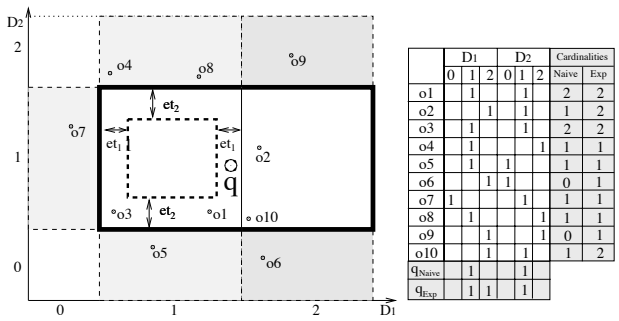
For the texture feature we have obtained EdgeHistogram and Homogeneous Texture descriptors with the MPEG-7 XM software. The Wavelet texture descriptor was provided by the City University of Hong Kong and Haralick Texture was locally implemented based on an ImageJ version [5].

2.2.3 Shape

We have used the MPEG-7 RegionShape descriptor also extracted also with the MPEG-7 XM reference sofware.

2.3 Local features

Keypoints or local interest points, feature is depicted by Scale Invariant Feature Transform (SIFT) descriptors [8] and provided by the City University of Hong Kong.



	D1			D2			Cardinalities	
	0	1	2	0	1	2	Naive	Exp
o1		1			1		2	2
o2			1		1		1	2
o3		1			1		2	2
o4		1				1	1	1
o5		1		1			1	1
o6			1	1			0	1
o7	1				1		1	1
o8		1				1	1	1
o9			1			1	0	1
o10			1		1		1	2
q_{Naive}		1			1			
q_{Exp}		1	1		1			

Figure 1: BitMatrix

2.3.1 Audio

We have processed the audio tracks of the shots to obtain a set of audio features with the help of the jAudio [9] software. The features are Spectral Centroid, Spectral Rollof Point, Spectral Flux, Compactness, Spectral Variability, Root Mean Square, Fraction of Low Energy Windows, Zero Crossings, Strongest Beat, Beat Sum, Strength of Strongest Beat, MFCC (Mel Frequency Cepstral Coefficient), LPC (Linear Predictive Coding) and Method of Moments.

3 The BitMatrix Indexing

For descriptor indexing, we use the BitMatrix [6], a method we have been developing for multimedia database applications. The BitMatrix method follows a data approximation approach in the spirit of the VA-File [10] and IGrid [4], partitioning each dimension D of the search space in k_D ranges. A partition of a dimension D is aset of ranges with chosen upper and lower bounds. The partitioning scheme is used to obtain bitmap signatures for the objects in the dataset, which are then arranged as lines in a matrix. For each dimension an object signature contains 1 for the range where the object belongs and 0 for the other ranges.

The search algorithm selects objects based on the cardinality (number of bits set to 1) of the bitwise AND between object and query signatures. Only the objects that obtain scores above an established cardinality threshold are then exhaustively analyzed and their exact distance is computed. Figure 1 illustrates a two-dimensional space with each dimension partitioned in three ranges.

The tradeoff between precision and speed is controlled by expanding the query regions and modifying the cardinality thresholds. In the example in Figure 1, expanding the query region along dimension D_1 changes the query signature from q_{Naive} to q_{Exp}. For the 10 objects with their signatures arranged in a BitMatrix, the cardinality columns $Naive$ and Exp contain the results of the bitwise AND with q_{Naive} and q_{Exp} respectively. If the cardinality threshold is

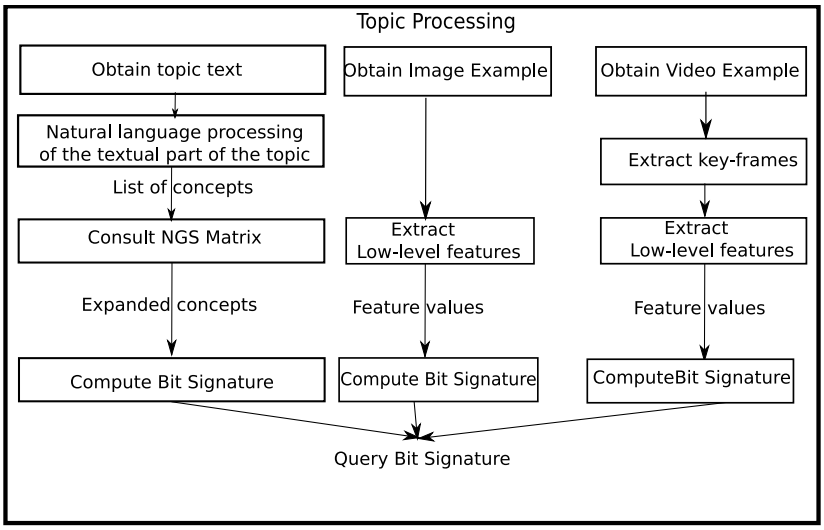


Figure 2: Topic Processing

set to 2, object O_2, which would be excluded with q_{Naive} is examined with q_{Exp}.

Several query objects can be easily combined with weighted bitwise AND/OR operations in order to obtain new hybrid query signatures. This is useful when answering complex queries that include several query objects or for relevance feedback scenarios when the user generally selects multiple relevant objects.

Our set of indexes for the TRECVID07 dataset consists of several BitMatrixes, one for each descriptor. The global dimensionality of the search space obtained by summing the partial dimensions introduced by each descriptor is 1628.

4 Topic Processing

The TRECVID topics contain a textual part and can cantain optionally image and video examples. Figure 2 shows the steps that we follow to translate the TRECVID topic into bit signatures, which are entries of the BitMatrix indexing system.

4.1 Natural Language Processing

The main goal of this task is to obtain a set of TRECVID concepts (39 in 2007) from a natural language question such as: "Find shots with one or more emergency vehicles in motion (e.g., ambulance, police car, fire truck, etc.)" (173-2006). This example results in the following list of Trecvid concepts: [car, walking, boat, disaster, airplane].

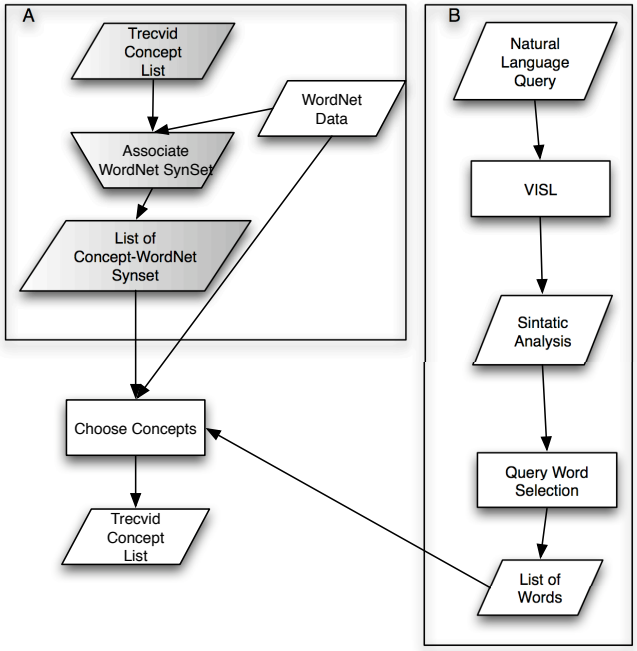


Figure 3: Processing Natural Language Queries

In order to map a natural language query into a subset of the 39 concepts we used the Wordnet [3] together with the natural language parser VISL (Visual Interactive Syntax Learning) [2, 1].

4.1.1 Representing the TRECVID Concept List

WordNet is lexical database for sets of synonyms called synsets with a set of semantic relations defined them. In our approach we only use the semantic relations between nouns namely: hyponym (kind of), meronym (part of), instance, and synonym. We use the WordNet Dabase in SWI-Prolog with the prolog files downloaded from the WordNet site.

Since each concept must correspond to a WordNet synset we had to translate (Figure 3A) the concepts into one or more synset, e.g. Concept "Vegetation" defined by sentence " Shots depicting natural or artificial greenery, vegetation woods, etc." is manually translated into the following synsets:

```
concept(vegetation,108436759)
concept(vegetation,100017222)
concept(vegetation,113153633)
```

identifying concepts of botany and flora in WordNet.

4.1.2 Obtaining a List of relevant words from a Query

The textual component of the topics are processed in order to extract a set of words that will be used to choose the TRECVID concepts associated to the topic (see Figure 3 B). For sentence 173, the resulting set of words is [emergency, vehicle, motion, ambulance, police car, truck]. This process has two setps. First we obtain the syntactic structure of the sentence using VISL, and then extract common and proper nouns.

Sentence 173-2006 above has the following sintactic structure:

```
coord('0173',1,1,w('Find','find','<mv>','V','IMP','@FS-COM',
'#1->0')).
coord('0173',1,2,w('shots','shot','N','P','NOM','@<ACC','#2->1')).
coord('0173',1,3,w('with','with','PRP','@<ADVL','#3->1')).
coord('0173',1,4,w('one=or=more','[one=or=more]',
'ADJ','POS','@>N','#4->8')).
coord('0173',1,5,w('emergency','emergency','N','S','NOM',
'@>N','#5->8')).
coord('0173',1,6,w('vehicles','vehicle','N','P','NOM','@>A',
'#6->7')).
coord('0173',1,7,w('in','in','ADJ','POS','@>N','#7->8')).
coord('0173',1,8,w('motion','motion','N','S','NOM',
'@P<','#8->3')).
coord('0173',1,9,w('e.g.','[e.g.]','ADV',
'@ADVL>','#10->17')).
coord('0173',1,10,w('ambulance','ambulance','N','S','NOM',
'@SUBJ>','#12->17')).
coord('0173',1,11,w('police','police','N','S','NOM',
'@>N','#14->15')).
coord('0173',1,12,w('car','car','N','S','NOM','@SUBJ>',
'#15->12')).
coord('0173',1,13,w('fire','fire','<mv>','V','IMP','@FS-<ADVL','fire',
'<mv>',  'V','PR','-3S','@FS-<ADVL',
'#17->1')).
coord('0173',1,14,w('truck','truck','N','S','NOM',
'@<ACC','#18->17')).
```

The process *Query Word Selection* in Figure 3 will look for: nouns, such as *truck* or *car*, nouns preceded by a *no* such as *no clouds*, compound nouns such as *police car* and names such as *Condoleezza Rice*, that are in Wordnet.

For this sentence the result is the list: [emergency, vehicle, motion, ambulance, police car, truck].

4.1.3 Choosing the concepts list

The last step is to pick each word and to compute the path from the word synset to all concept synsets. Finally we have to choose the best scored concepts, using an heuristic based on the computed distances.

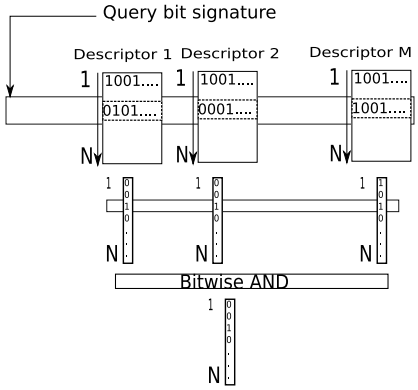


Figure 4: Answer computation

4.2 Concept Expansion

The sets of concepts obtained from the NLP analysis of the topic text is not influenced by the video collection. It is interesting to find the distances between concepts in the context of the TRECVID dataset. To obtain them, the high-level annotations have been processesed (an off-line process), by computing a distance matrix between the 39 LSCOM-Lite concepts. We have applied the concept of Google Similarity Distance (GSD) [7] to compute a distance between concepts dependent on the TRECVID dataset. With this approach, we have found for example that “Building" is similar to “Court" and “Flag-US" and therefore could be also used in queries that ask for buildings. Figure 2 shows that at query time, after the NLP-based translation of the topics into trecvid concepts the GSD matrix is consulted. The result is the expanded list of TRECVID concepts.

4.3 Examples Processing

The topics include examples in two modalities: image and video. As our system uses mostly image-based features the query examples have been processed to obtain image features. They were obtained from the image examples provided with the topics and from the video examples by manually extracting representative frames. The feature values are then obtained by extracting the specified descriptors (See Section 2) .

Figure 2 illustrates the phases of translating a TRECVID topics into bit signatures. First the expanded concepts list are translated into bit signatures. The concept signature is a bit sequence of length 39 where each position indicates whether the concept is present or not. Then low-level features for the image examples and for the key-frames of the video examples are obtained. The concept lists and the feature values are then used to obtain the query's bit signature by using weighted AND/OR bitwise operations.

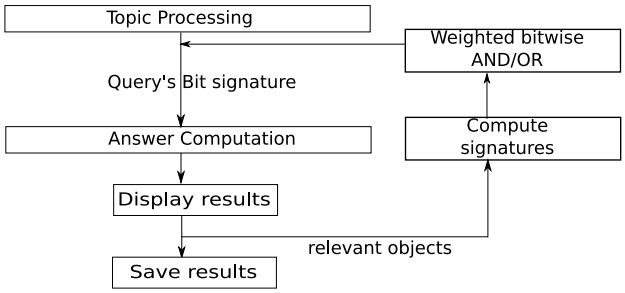


Figure 5: The Search Process

5 The Search Process

At search time we have the BitMatrix structures for the low and high-level features and the query bit signatures obtained from the topics. The process is started from the search interface by choosing the topic number. The corresponding query bit signature is sent to the BitMatrix for computing the answer.

5.1 The Search Interface

The search interface is a available as a web application. It allows the user to choose custom search scenarios: natural language query, query by example or, for the TRECVID setup, query by topic number. The results are displayed on a wide scrollable surface depicted in Figure 6 and can be marked for subsequent search iterations, relevance feedback. The interface allows each iteration to be configured by choosing custom combinations of descriptors. For example the search can start with color and texture descriptors and switch to audio or shape descriptors.

5.2 Answer Computation

The query bit signature is the input of the BitMatrix search system. Figure 4 shows separate matrixes for each descriptor (Concepts, ColorLayout,ColorStructure, ColorMoments, Wavelet, ... SIFT) in order to let the user choose a combination of them. The search method presented in Section 3 is applied for each descriptor in the set and results lists are obtained. Each list represents approximate similar objects with respect to the corresponding descriptor. The lists contain the scores, which are the cardinalities computed after bitwise AND between the shots and the query bit signatures. By defining a cardinality treshold we obtain binary lists that contain 1 if a shot has a higher cardinality than the treshold and 0 otherwise. The aggregation at this stage can be done either using the scores lists, or the binary lists.

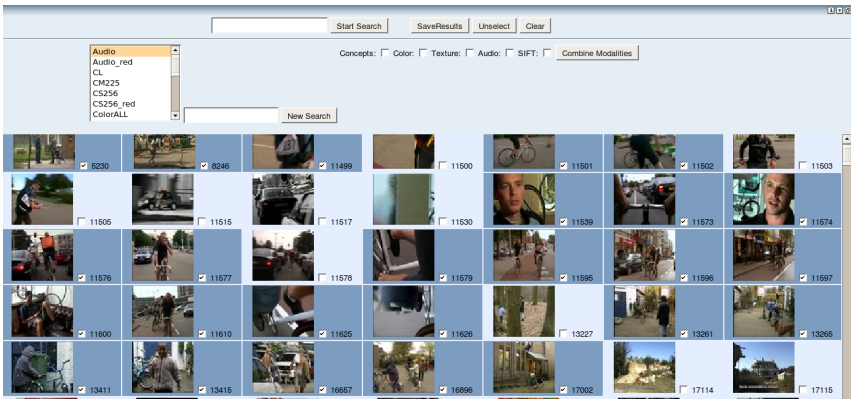


Figure 6: The Search Interface

5.3 Relevance Feedback

Figure 5 shows the whole search process. After running the BitMatrix search procedure, the results are displayed on the search interface as depicted in Figure 6. The user has the possibility of selecting relevant shots that may be combined with the initial examples to form new query bit signatures. The required setup for subsequent search iterations consist of new query bit signatures and the combination of descriptors that may be changed at every step.

6 Results and Conclusion

Figure 7 illustrates the results for our automatic run. It performs above the median, in spite of not using any classifier or any other knowledge sources except the translation of the query into LSCOM-Lite concepts. The database indexing perspective that we have chosen for the search task, allows us to perform quick relevance feedback steps using an approximate similarity metric based on bitwise operations. By using a large number of descriptors we can see many facets of “similar" objects (from different perspectives) and aggregate partial lists in flexible ways. It is easy to gather low and high-level features in our indexing structure. We can therefore experiment novel ranking strategies.

We have used only one key-frame per shot. Improvements can be obtained by using more frames from each shot. Incorporating descriptors for the whole shot (such as GroupOfFrames/GroupOfPictures, MotionActivity) and not only for a frame should also be tested.

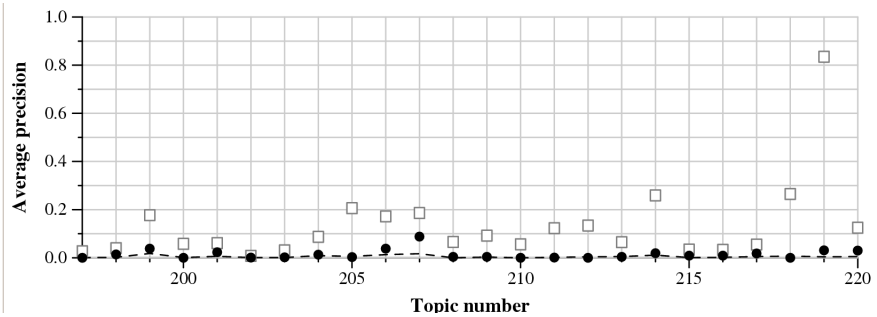


Figure 7: Average Precision

References

[1] Visl: web application, September. `http://beta.visl.sdu.dk/visl/en/parsing/automatic/parse.php` em 2007-08-30.

[2] Visl: website. `http://visl.sdu.dk/` em 2007-09-15.

[3] Wordnet: Official website. `http://wordnet.princeton.edu/` em 2007-09-20.

[4] Charu C. Aggarwal and Philip S. Yu. The IGrid index: reversing the dimensionality curse for similarity indexing in high dimensional space. In *KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 119–129, New York, NY, USA, 2000. ACM Press.

[5] Werner Bailer. Writing ImageJ PlugIns - A Tutorial, July 2006.

[6] Catalin Calistru, Cristina Ribeiro, and Gabriel David. Multidimensional Descriptor Indexing: Exploring the BitMatrix. In *CIVR*, pages 401–410, 2006.

[7] Rudi Cilibrasi and Paul M. B. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19:370, 2007.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision*, volume 20, pages 91–110, 2003.

[9] Daniel McEnnis, Cory McKay, and Ichiro Fujinaga. jAudio: Additions and Improvements. pages 385–386, October 2006.

[10] Roger Weber, Hans-Jörg Schek, and Stephen Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proc. 24th Int. Conf. Very Large Data Bases, VLDB*, pages 194–205, 24–27 1998.